

```

finally {tokPodatkov.close(); } // datoteki zapremo

// priprava na branje datoteke
FileInputStream vhodniTok = new FileInputStream("skladisce.dat");
DataInputStream podatki = new DataInputStream(vhodniTok);

try //branje treh celih števil iz datoteke in izpis na zaslon
    {d = podatki.readInt(); System.out.println(d);
      System.out.println(podatki.readInt()); //krajše
      System.out.println(podatki.readInt());
      System.out.println(podatki.readInt());
      System.out.println(podatki.readInt());
    }

catch(Exception e)
    {System.out.println(e.getMessage()); }
  finally{podatki.close(); }
}

```

Vsebino si ogledamo v šestnajstiški obliki¹.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
000000	00	00	00	00	00	00	00	05	FF	FF	FF	FB	00	00	00	1D
000010	00	00	00	FF												

- Spomnimo se: vsak dvomestni šestnajstiški zapis predstavlja en bajt vsebine.
- Vidimo, da je bilo vsako od petih shranjenih števil zapismo s 4 bajti. (Podatki tipa *int*!)
- V začetku prepoznamo števili 0 in 5: 00 00 00 00 00 in 00 00 00 00 05.
- Naslednje število, FF FF FF FF FB, je negativno, saj ima najtežji bit (MSB) vrednost 1. Torej ga moramo odkodirati kot negativno število, zapisano z dvojiškim komplementom². Dobimo -5.
- Sledita še dve pozitivni celi števili, 29 in 255.

Na podoben način lahko v datoteko shranimo *logične vrednosti* (tipa *boolean*). Uporabimo ukaz *writeBoolean(true)* oz. *writeBoolean(false)*. Kako so logične vrednosti zapisane (kodirane) v datoteki?

Ustvarimo datoteko s takšnimi podatki in s *hex-viewerjem* ugotovimo, da sta vrednosti *true* in *false* v datoteki zapisani (šestnajstiško) kot 01 (*true*) oz. 00 (*false*). Čeprav vsaka logična vrednost nosi informacijo velikosti en **bit**, je v datoteki zanj 'porabljen' cel bajt.

Podobno si lahko v datotekah "ogledamo" tudi zapise realnih števil, podatke drugih tipov in zapise različnih podatkovnih struktur, na primer tabel.

¹ uporabimo lahko enega od brezplačnih *hexviewer*-jev, kot v našem primeru

² Od FF FF FF FB odštejemo 1, dobimo FF FF FF FA, nato invertiramo vse bite in dobimo 00 00 00 05. Število je torej -5.